# Метод обучения персептрона

В качестве научного предмета искусственные нейронные сети впервые заявили о себе в 40-е годы. Стремясь воспроизвести функции человеческого мозга, исследователи создали простые аппаратные (а позже программные) модели биологического нейрона и системы его соединений. Когда нейрофизиологи достигли более глубокого понимания нервной системы человека, эти ранние попытки стали восприниматься как весьма грубые аппроксимации. Тем не менее, на этом пути были достигнуты впечатляющие результаты, стимулировавшие дальнейшие исследования, приведшие к созданию более изощренных сетей. Первое систематическое изучение искусственных нейронных сетей было предпринято Маккалокком и Питтсом в 1943 г.

Позднее в работе они исследовали сетевые парадигмы для распознавания изображений, подвергаемых сдвигам и поворотам. Простая нейронная модель, показанная на рис. 2.1, использовалась в большей части их работы. Элемент **S** умножает каждый вход х на вес w и суммирует взвешенные входы. Если эта сумма больше заданного порогового значения, выход равен единице, в противном случае - нулю. Эти системы (и множество им подобных) получили название персептронов. Они состоят из одного слоя искусственных нейронов, соединенных с помощью весовых коэффициентов с множеством входов, хотя в принципе описываются и более сложные системы.

В 60-е годы персептроны вызвали большой интерес и оптимизм. Розенблатт доказал замечательную теорему об обучении персептронов. Уидроу дал ряд убедительных демонстраций систем персептронного типа, и исследователи во всем мире стремились изучить возможности этих систем. Первоначальная эйфория сменилась разочарованием, когда оказалось, что персептроны не способны обучиться решению ряда простых задач. Минский строго проанализировал эту проблему и показал, что имеются жесткие ограничения на то, что могут выполнять однослойные персептроны, и, следовательно, на то, чему они могут обучаться. Так как в то время методы обучения многослойных сетей не были известны, исследователи перешли в более многообещающие области, и исследования в области нейронных сетей пришли в упадок. Недавнее открытие методов обучения многослойных сетей в большей степени, чем какой-либо иной фактор, повлияло на возрождение интереса и исследовательских усилий

## Обучение персептрона

Способность искусственных нейронных сетей обучаться является их наиболее интригующим свойством. Подобно биологическим системам, которые они моделируют, эти нейронные сети сами моделируют себя в результате попыток достичь лучшей модели поведения.

Персептрон обучают, подавая множество образов по одному на его вход, и подстраивая веса до тех пор, пока для всех образов не будет достигнут требуемый выход. Допустим, что входные образы нанесены на демонстрационные карты. Каждая карта разбита на квадраты и от каждого квадрата на персептрон подается вход. Если в квадрате имеется линия, то от него подается единица, в противном случае - ноль. Множество квадратов на карте задает, таким образом, множество нулей и единиц, которое и подается на входы персептрона. Цель состоит в том, чтобы научить персептрон включать индикатор при подаче на него множества входов, задающих нечетное число, и не включать в случае четного.

На рис. 2.1 показана такая персептронная конфигурация. Допустим, что вектор **Х** является образом распознаваемой демонстрационной карты. Каждая компонента (квадрат) **Х** - (x1, x2, :, xn) - умножается на соответствующую компоненту вектора весов **W** - (w1, w2, ..., wn).

Эти произведения суммируются. Если сумма превышает порог P, то выход нейрона **Y** равен единице (индикатор зажигается), в противном случае он - ноль. Как известно из предыдущей главы, эта операция компактно записывается в векторной форме как **Y** = **XW**, а после нее следует пороговая операция. Для обучения сети образ **Х** подается на вход и вычисляется выход **Y**. Если **Y** правилен, то ничего не меняется. Однако если выход неправилен, то веса, присоединенные к входам, усиливающим ошибочный результат, модифицируются, чтобы уменьшить ошибку. Чтобы увидеть, как это осуществляется, допустим, что демонстрационная карта с цифрой 3 подана на вход и выход **Y** равен 1 (показывая нечетность). Так как это правильный ответ, то веса не изменяются. Если, однако, на вход подается карта с номером 4 и выход **Y** равен единице (нечетный), то веса, присоединенные к единичным входам, должны быть уменьшены, так как они стремятся дать неверный результат. Аналогично, если карта с номером 3 дает нулевой выход, то веса, присоединенные к единичным входам, должны быть увеличены, чтобы скорректировать ошибку.

**Этот метод обучения может быть подытожен следующим образом:**

1. Подать входной образ и вычислить **Y**.
   1. Если выход правильный, то перейти на шаг 1;
   2. Если выход неправильный и равен нулю, то добавить все входы к соответствующим им весам; или
   3. Если выход неправильный и равен единице, то вычесть каждый вход из соответствующего ему веса.
2. Перейти на шаг 1.

За конечное число шагов сеть научится разделять карты на четные и нечетные, при условии, что множество цифр линейно разделимо. Это значит, что для всех нечетных карт выход будет больше порога, а для всех четных - меньше. Отметим, что это обучение глобально, т. е. сеть обучается на всем множестве карт. Возникает вопрос о том, как это множество должно предъявляться, чтобы минимизировать время обучения. Должны ли элементы множества предъявляться последовательно друг за другом или карты следует выбирать случайно? Несложная теория служит здесь путеводителем.

## Дельта-правило

Важное обобщение алгоритма обучения персептрона, называемое дельта-правилом, переносит этот метод на непрерывные входы и выходы. Чтобы понять, как оно было получено, шаг 2 алгоритма обучения персептрона может быть сформулирован в обобщенной форме с помощью введения величины *delta*, которая равна разности между требуемым или целевым выходом T и реальным выходом Y

*delta* = (T - Y).

Случай, когда *delta* = 0, соответствует шагу 2a, когда выход правилен и в сети ничего не изменяется. Шаг 2b соответствует случаю *delta* > 0, а шаг 2c случаю *delta* < 0. В любом из этих случаев персептронный алгоритм обучения сохраняется, если *delta* умножается на величину каждого входа хi и это произведение добавляется к соответствующему весу. С целью обобщения вводится коэффициент "скорости обучения" n), который умножается на *delta*хi, что позволяет управлять средней величиной изменения весов. В алгебраической форме записи

Di = n*delta*xi,

w(n+1) = w(n) + Di,

где Di - коррекция, связанная с i-м входом хi; wi(n+1) - значение веса i после коррекции; wi(n) -значение веса i до коррекции. Дельта-правило модифицирует веса в соответствии с требуемым и действительным значениями выхода каждой полярности как для непрерывных, так и для бинарных входов и выходов. Эти свойства открыли множество новых приложений.

## Трудности с алгоритмом обучения персептрона

Может оказаться затруднительным определить, выполнено ли условие разделимости для конкретного обучающего множества. Кроме того, во многих встречающихся на практике ситуациях входы часто меняются во времени и могут быть разделимы в один момент времени и неразделимы в другой. В доказательстве алгоритма обучения персептрона ничего не говорится также о том, сколько шагов требуется для обучения сети. Мало утешительного в знании того, что обучение закончится за конечное число шагов, если необходимое для этого время сравнимо с геологической эпохой.

Кроме того, не доказано, что персептронный алгоритм обучения более быстр по сравнению с простым перебором всех возможных значений весов, и в некоторых случаях этот примитивный подход может оказаться лучше. На эти вопросы никогда не находилось удовлетворительного ответа, они относятся к природе обучающего материала. В различной форме они возникают в последующих главах, где рассматриваются другие сетевые парадигмы. Ответы для современных сетей, как правило, не более удовлетворительны, чем для персептрона. Эти проблемы являются важной областью современных исследований.